

# EECS 214 Practice Quiz 2

---

## Question 1

Consider the following sort algorithm:

```
Sort(array)
  T = make an empty red/black tree
  for each element e in array
    add e to T
  for i=0 to array.length-1
    e = T.FindMinimum
    T.Remove(e)
    array[i] = e
```

What is the time complexity (i.e. O of what?) of this sort?

## Question 2

Suppose you are given a hash table using chaining (i.e. linked lists) and it uses the following awful hash function:

$$h(k) = 0$$

What is the complexity (i.e. O of what?) of the following algorithm:

```
for i=1 to n
  hashtable.Store(i, 0)
```

Assume that the hash table has to check whether the key  $i$  is already present when doing the store operation.

### Question 3

Here's a random algorithm that operates on an array. Yes, I know it doesn't do anything useful, but tell me what its time complexity is anyway (i.e. is it linear, quadratic, log n, or what?).

```
BlaBlaBla(array, start, end)
  if (start!=end)
    for each i between start and end
      sum += array[i]
    BlaBlaBla(array, start, (start+end)/2)
    BlaBlaBla(array, (start+end)/2, end)
```

### Question 4

Write a recursive algorithm to count the number of nodes in a connected graph. You may assume any graph representation you find convenient, however, if you use the array of adjacency lists representation, you can't cheat and just check the length of the array :-)

### Question 5

Suppose you're given an undirected graph represented using the array-of-adjacency-lists representation used in the book. Give an algorithm for finding the distance **from a given node to every other node in the graph**. If a node is not reachable from the start node, its distance should be marked as infinite. Be sure it will not loop infinitely if the graph has cycles.

### Question 6

Suppose you're given a **weighted**, undirected graph represented using an adjacency matrix representation (i.e. a matrix of weights where the weight is  $\infty$  if the nodes are

unconnected). Give an algorithm for finding the distance **from every node to every other node** in the graph. If two nodes are unconnected, their distance should be marked as infinite. Your algorithm should run in  $O(V^3)$  time.

## Question 7

What is the worst-case asymptotic time complexity of Dijkstra's algorithm, assuming it is implemented using a binary heap?

## Question 8

Explain the differences between a binary search tree and a binary heap.